Globus Quick Start Guide

Globus Software Version 1.1.3 *September 2000*

The Globus Project is a community effort, led by Argonne National Laboratory and the University of Southern California's Information Sciences Institute. Globus is developing the basic software infrastructure for computations that integrate geographically distributed computational and information resources.

This document was produced as a community effort by the Information Power Grid (IPG) group at NASA Ames Research Center, Code IN; Argonne National Laboratory; the National Computational Science Alliance; and the University of Southern California's Information Sciences Institute.

Please report errors in this document to documentation@globus.org

Contents

Summary of New Features in Globus 1.1.3

Release 1.1.3 of the Globus Toolkit provides a more scalable and flexible Grid Information Service (GIS), also known as MDS, and streamlined interfaces for the Globus Resource Allocation and Management (GRAM) tools. Additionally, release 1.1.3 includes a new version of the globus_io library with significantly improved performance. The following differences affect the user interface:

- `grid-cert-request`: Several organizations now have their own Certificate Authority (CA). Ask your local Globus administrator for information on obtaining a Certificate. See page 8.
- Resource contact strings for "fork" services are no longer routinely suffixed with `/jobmanager-fork`. If you do not specify a jobmanager in your `globusrun`, `globus-job-run`, or `globus-job-submit` command, your job will be submitted to the default jobmanager on the remote system, which is usually fork. If you want to specify a non-default jobmanager, do so by appending `/jobmanager` and the name of the job scheduler, such as `/jobmanager-lsf` or `/jobmanager-pbs`, to the hostname. See page 15.
- GIS/MDS: Globus group no longer runs a centralized MDS server. That means that each site optionally will have its own organization server, which is used for any searching commands. See page 20.

Known Deficiencies

IRIX

When using the Globus Toolkit on IRIX systems, the `sproc_mpi` and `pthreads_mpi` libraries are not built properly. We do not recommend using these libraries on IRIX at this time.

Chapter 1 Extremely Quick Start

Here are the briefest instructions for getting started with Globus Version 1.1.3.

1. Get accounts on Globus-enabled machines.
2. Login to a Globus-enabled machine.
3. Setup your environment and path, if necessary.
4. Ask your local Globus administrator for information on obtaining a Certificate. Some grids now have their own Certificate Authority (CA). If your certificate is to be obtained from the Globus CA, enter `grid-cert-request` on your Unix command line. Follow instructions. Be sure to send your subject name from the `grid-cert-request` response to the system administrators of all your accounts on which you plan to use Globus.
5. Wait 2 days or fewer to receive your signed certificate via email. Follow the instructions in the email for how to save your certificate.
6. Enter `globus-setup-test`. If that gives you "Success!" then:
7. Enter `grid-proxy-init`
8. Enter `globus-job-run <hostname> /bin/echo "Hello World."`
9. Use `<globus command> -help` for information on Globus commands.

Chapter 2 Introduction

Grids are super Internets for high-performance computing: worldwide collections of high-end resources—such as supercomputers, storage, advanced instruments, and immersive environments. These resources and their users are often separated by great distances and connected by high-speed networks.

The Globus development team has created a set of underlying Grid services and a software toolkit for using the geographically distributed resources on Grids.

This chapter contains background information only, no instructions. This manual assumes you know Unix. The "%" sign represents the Unix prompt in examples.

What are Grids?

Grids bring together geographically and organizationally dispersed computational resources, such as CPUs, storage systems, communication systems, real-time data sources and instruments, and human collaborators. If you are developing an application that requires geographically distributed high-end resources, you will want to know about Grids. Grids are new—many of the enabling technologies have not yet been invented. If you are reading this in the early 2000s, you are one of the Grid pioneers.

Current testbed grids will eventually become "the Grid." In much the same way that the nationwide electric power grid connects sources of electricity, the information Grid will connect sources and users of high-performance computing cycles, allowing these cycles to be generated in one place and used in another. Remote collaborators will work together on large-scale projects. Scientists will be able to access extremely data-intensive instruments from across the country, in real time.

The goal of the Grid community is to provide dependable, consistent, pervasive access to high-end resources.

What is Globus?

The Globus software toolkit facilitates the creation of usable Grids, enabling high-speed coupling of people, computers, databases, and instruments. With Globus, you can run your gigabyte-per-time-step dataset job on two or more high-performance machines at the same time, even though the machines might be located far apart and owned by different organizations. Globus software helps scientists deal with very large datasets and complex remote collaborations.

Globus software is used for large distributed computational jobs, remote instrumentation, remote data transfer, and shared immersive spaces. For example,

- Globus used in live runs at the world's brightest x-ray source : http://www.mcs.anl.gov/~laszewsk/xray/cmt/gallery/
- Distributed supercomputing, smart instruments, teleimmersive applications: http://www.globus.org/overview/applications.html

Globus Features

Globus is designed to offer features such as uniform access to distributed resources with diverse scheduling mechanisms; information service for resource publication, discovery, and selection; API and command-line tools for remote file management, staging of executables and data; and enhanced performance through multiple communication protocols.

Testbeds

Grids are a community effort. University, commercial, and government computer science communities contribute to the development of Grids. Several significant testbeds are up and running. You can obtain accounts by sending email to the addresses given below. Include your who-why-where: full name, username if you have one on that system, company or affiliation, and brief description of work.

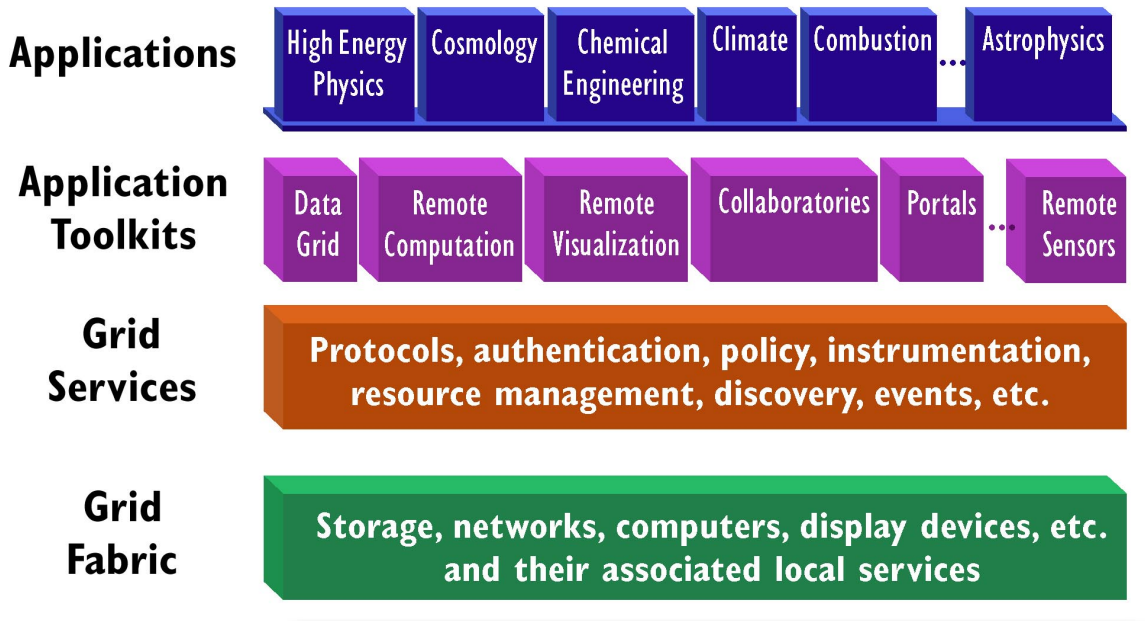
Bear in mind that these Grids are works-in-progress; things still break.

Each of these Grids uses Globus services to provide uniform access to a large collection of resources for a particular Grid community. Work is ongoing to enable interoperation of these various testbeds; however, most likely you will be using one particular testbed. There are other testbeds in progress.

- GUSTO—this was the original testbed for the Globus software.
- IPG, the Information Power Grid—this is NASA's testbed for Grid research. See http://www.ipg.nasa.gov To apply for accounts, send email to the accounts@nas.nasa.gov alias.
- National Technology Grid—this is the testbed created by the National Computational Science Alliance, led by NCSA. See http://www.ncsa.uiuc.edu/alliance/g-force/. To apply for accounts, send email to the allocations@ncsa.uiuc.edu alias.
- The National Partnership for Advanced Computational Infrastructure (NPACI), led by the San Diego Supercomputing Center, operates a testbed. See http://www.npaci.edu/.

Four Levels of the Grid

The Integrated Grid Architecture



Grid Fabric: Layer One

The *fabric* of the Grid comprises the underlying systems, computers, operating systems, networks, storage systems, and routers—the building blocks.

Grid Services: Layer Two

Grid services *integrate* the components of the Grid fabric. Examples of the services that are provided by Globus:

GRAM

The Globus Resource Allocation Manager, GRAM, is a basic library service that provides capabilities to do remote-submission job start up. GRAM unites Grid machines, providing a common user interface so that you can submit a job to multiple machines on the Grid fabric. GRAM is a general, ubiquitous service, with specific application toolkit commands built on top of it.

GIS

The Grid Information Service, GIS, formerly known as the Metacomputing Directory Service, MDS, provides information service. You query GIS to discover the properties of the machines, computers and networks that you want to use: how many processors are available at this moment? What bandwidth is provided? Is the storage on tape or disk? Is the visualization device an immersive desk or CAVE? Using an LDAP (Lightweight Directory Access Protocol) server, GIS provides middleware information in a common interface to put a unifying picture on top of disparate equipment.

GSI

The Grid Security Infrastructure, GSI, is a library for providing generic security services for applications that will be run on the Grid. Application programmers use the `gss-api` library for adding authentication to a program. GSI provides programs, such as `grid-proxy-init`, to facilitate login to a variety of sites, while each site has its own flavor of security measures. That is, on the fabric layer, the various machines you want to use might be governed by disparate security policies; GSI provides a means of simplifying multiple remote logins.

Application Toolkits: Layer Three

Application toolkits use Grid Services to provide higher-level capabilities, often targeted to specific classes of application.

For example, the Globus development team has created a set of Grid service tools and a toolkit of programs for running remotely distributed jobs. These include remote job submission commands (`globusrun`, `globus-job-submit`, `globus-job-run`), built on top of the GRAM service, and MPICH-G, a Grid-enabled implementation of the Message Passing Interface (MPI).

A number of groups are also developing a range of other toolkits such as support for distributed management of large datasets, collaborative visualization, and online instrumentation. These are not discussed in this document.

Specific Applications: Layer Four

A rich variety of applications have been developed that build on services provided by the three layers just described. For example, OVERFLOW is a thin-layer Navier-Stokes flow solver for structured grids that has been modified to operate on multiple supercomputers via the use of MPICH-G.

Other application areas being targeted to the Grid include high-energy physics, cosmology, chemical engineering, climate, combustion, and astrobiology.

Purpose of This Document

This *Globus Quick Start Guide* tells you how to get started using Globus Grid services and the tools built on top of those services to build applications to exploit geographically distributed resources. The instructions herein are brief and do not take you much past “Hello World,” but do provide the essentials required to get started.

Support and Usage Help

Address questions to your local system administrator. Also refer to the Bibliography and Reference on page 28. For usage information about any Globus command, type the command with the `-help` or `–usage` option:

```
% <globus-command-name>  -help
% <globus-command-name>  -usage
```

All programs have a man page with more elaborate explanations than provided by `-help` or `–usage`. See http://www.globus.org/v1.1/programs/ for information.

About this Guide

Audience

This guide is intended for novice Globus users.

Assumptions

Before you start using Globus, you will need to fulfill these basic requirements.

Know Basic Unix

This manual assumes that you know at least the fundamentals of Unix. You can find good Unix information at http://www.ugu.com/sui/ugu/show?help.beginners.

Globus Is Installed

If you are planning to use the resources on the existing Grid testbeds, you probably will find Globus already installed. If you want to add your resources to a testbed, ask your system administrator to see http://www.globus.org/software/installation.html.

Globus provides distinct server and client software. Server software is intended for machines to which remote access is desired; in current testbeds, server software is often installed only on high-end resources. Client software allows users to access remote servers; it is installed on workstations, laptops, and other desktop machines. If you want to run Globus from your desktop machine, you can login to a Globus machine, or install Globus *client* software on your machine.

Organization and Conventions

The chapters are presented in step-by-step sequence. Only the bare essentials are included. A glossary defines terms. Globus commands start with globus or grid. “Resources” refers to computers, instruments, or other machines (as opposed to data)

Conventions used in this guide:

%	Unix prompt; do not enter
$	precedes a variable
<*italics*>	substitute your own information. For example, <*src-dir*> means enter in your source directory pathname. Italics are also used within plain text for emphasis.
`[ ]`	square brackets indicate parameters that sometimes may be omitted.
\	indicates that a command line continues

How to Use This Manual Online

If you are reading the PDF version online, use the button marked with a T or ABC to make text selectable; then you can copy the examples and paste them into your Unix command line. Be careful not to copy the backslashes (\) into the middle of your command lines.

Chapter 3 Globus Certificate

In this chapter, you obtain a Globus Certificate to have authenticate-once access to all of the machines you want to use on the Globus testbed. Your Globus Certificate is like a passport; it establishes your identity on the Grid. This is what you do:

- Obtain accounts on geographically distributed machines, and path names to Globus tools.
- Check, and if necessary, set your GLOBUS_INSTALL_PATH environment variable and your path to Globus tools.
- Initiate the `grid-cert-request` command.
- Email all system administrators (for those resources that you wish to access via Globus tools) with your subject name—your unique Grid identifier.
- Save your `usercert.pem` and change the permissions as instructed in the `grid-cert-request` response.

NEW: some Grids have their own Certificate Authority, and a different way of applying for a certificate. Ask your local Globus administrator.

Why You Want a Certificate

With a Globus certificate, you only have to enter your password once per Globus session. The Globus certificate is like a passport. You take the time to establish your identity to the proper authorities, and then you are issued a certificate and private key. The certificate and key give you geographically distributed access. You do not have to reestablish your identity at each "border."

Local Accounts

Before using Globus, you will need to get accounts on all of the Grid computers you plan to use. Talk to your local system administrator, if your local system is part of the Grid.

You should have one computer, running Globus client software, that you consider your home/local machine. Get your certificate from your local home machine. That is, the Certificate Authority (CA) will consider your home machine to be the machine from which you apply for your certificate.

Set Your Environment and Path

This is a critical step! Perhaps your kind and thoughtful system administrators have done it for you. To find out, type:

```
% which globus-job-submit
```

If you get a response that looks something like:

```
/software/globus-1.1.3/release/tools/mips-sgi-irix6.5/bin/globus-job-submit
```

your path and environment variable probably have been set for you. If not, you will have to work with your system administrators to get it right. You need to set the GLOBUS_INSTALL_PATH environment variable, and add the Globus tools directory to your default search path. This path varies per system.

NOTE: Your previous Globus settings may create havoc. Get rid of them.

You can use the following commands within your Unix shell, or add them to your own dot files. Substitute `<path>` with the path to where Globus is installed on your system:

```
(csh, tcsh):
           setenv GLOBUS_INSTALL_PATH <path>
           source $GLOBUS_INSTALL_PATH/etc/globus-user-setup.csh
(bash, ksh):
           export GLOBUS_INSTALL_PATH=<path>
           . $GLOBUS_INSTALL_PATH/etc/globus-user-setup.sh
(sh)
           GLOBUS_INSTALL_PATH=<path>
           export GLOBUS_INSTALL_PATH
           . $GLOBUS_INSTALL_PATH/etc/globus-user-setup.sh
```

Look at your path by using the echo command. Check to see that there is no older Globus path preceding the new one; if there is an older path, remove it or move it to after the new one. Failure to fix this will give you older versions of Globus commands.

```
% echo $PATH
% echo $GLOBUS_INSTALL_PATH
```

Globus Certification: Using grid-cert-request

Globus provides authenticate-once capability: a common certificate/key security method that allows single sign-on anywhere you have accounts on the Grid. In other words, after you have obtained a Globus certificate, one login gives you access to all the Grid resources on which you have accounts. This is an important feature, because it eliminates the nuisance of reentering your login and password every time you utilize another resource.

Version 1.1.3 Note: This section contains instructions for obtaining a certificate from the Globus Certificate Authority (CA). **Several organizations now have their own Certificate Authority (CA). Ask your local Globus administrator for information on obtaining a Certificate.**

Easy Steps to Certification

Step 1—Login

Login to your home computer in the Grid. See “Local Accounts” on page 7 for clarification.

Step 2—Executing grid-cert-request

Execute the `grid-cert-request` command.

```
% grid-cert-request
```

Step 3—Entering a Passphrase

Follow all instructions presented by the `grid-cert-request` program. When asked to enter a passphrase, use at least 8 characters, including mixed-case alpha characters and at least one numeral or punctuation mark.

NOTE: You are the only person or thing that knows your passphrase. Do not forget it! No one can retrieve it for you later. If you forget your passphrase, you will have to get a whole new certificate.

Step 4—Sending email to a Certificate Authority (CA)

An important part of the `grid-cert-request` command output is as follows:

```
Please e-mail the certificate request to the Globus CA
cat /u/your_user_name/.globus/ usercert_request.pem | mail
ca@globus.org
```

The `grid-cert-request` creates a dot-globus directory (`.globus`) in your home directory.. Just copy the line starting with cat and paste it into a Unix command line; this will cause your `usercert_request.pem` file to be mailed to the CA (*which* CA will depend on your organization; follow the instructions you are given). You should get your certificate within two days. Send the mail from a machine on which you can receive or forward email. If you send from a machine that cannot receive mail, you will get no response. If you get no response, send a note, not a new certificate request, to ca@globus.org or your certificate authority.

Remember that in Globus version 1.1.3, some organizations have their own Certificate Authority (CA). The response to your `grid-cert –request` will tell you where to send your email.

Step 5—Sending email Again (and Again)

When you receive email from the Globus Certification Authority in response to your `grid-cert-request` email, read the response carefully and follow the directions therein.

Have your unique Grid identifier added to the grid-mapfile at each resource (machine) you intend to use. To do this, email your *subject name* to your Globus Grid system administrators. Your *subject name* is near the top of the `grid-cert-request` email response. For example:

```
Enclosed is your Globus Certificate for:
/O=Grid/O=Globus/OU=sdsc.edu/CN=Lucy Mcgillicutty
```

Email your Certificate Subject (starting with /O=) to the system administrator for *each* machine you plan to use (i.e., all the Grid machines on which you have accounts). If you do

not know the system administrator's address, send email to the same address you used for getting the account.

Step 6—Saving Your Information

Save the usercert.pem in your dot-globus directory. Here is an example of part of one `grid-cert-request` output:

```
Save this e-mail message into the following file.
/u/mcgillicutty/.globus/usercert.pem
```

Use the exact path in *your* email message. If you are asked if you want to overwrite usercert.pem, enter *yes*. This file contains your signed public key, now saved where Globus knows to find it.

Step 7—Changing Permissions

Change permissions on .pem files in your dot-globus directory. Enter **`ls -l`** to list your files. Use the `chmod` command to change permissions:

```
% chmod 400 userkey.pem
% chmod 444 usercert.pem
```

Checking Version

You will run into trouble with the commands in this book if you are not running Globus 1.1.x. To check, enter:

```
% globus-version
```

If you are not getting a valid version number such as 1.1.3, see Setting Environment and Path, page 7.

Ready to run

You are done. Your system administrator will let you know when your name has been added (by humans) to the grid-mapfile(s) on your Grid resources. See the next section for how to check if you are in grid-mapfile. After you have successfully completed the certificate test below you need to get a grid-proxy, which is easy and instant. See page 11. Don't forget that passphrase. As you run into bumps in the road, remember that you are a Grid pioneer. Do not expect all the roads to be paved. Grids do not yet run smoothly.

Certificate Test

On a Grid computer, enter:

```
% globus-setup-test
```

If you get

```
Authentication test ……Failure!
```

your unique identifier has not yet been added to the grid-mapfile, or there may be a system problem. Talk to the system administrator.

Chapter 4 Proxy Credential

In this chapter you obtain a Globus proxy by typing `grid-proxy-init`. The proxy gives you authenticate-once capability for 12 hours (default). While your proxy is active, you can log into any Grid resource without reentering your passphrase.

The command `grid-proxy-info` displays contents/status of your proxy certificate.

Obtaining a Proxy Credential

When you have obtained accounts on some Grid testbed machines and followed the `grid-cert-request` procedures, you are ready to get a proxy. The Globus proxy is based on a so-called “public key” security method that allows single sign-on anywhere a user has accounts on a Grid. One proxy is good for 12 hours (default). When you start another session on another day, you get a new proxy.

Using the grid-proxy-init Command

At your system prompt, enter `grid-proxy-init` and you are prompted for your Globus passphrase.

```
% grid-proxy-init
```

prints:

```
Enter PEM pass phrase:
.      ..........................+++++
.      ..................+++++
%
```

You can set options with command line arguments to `grid-proxy-init`; type the command with the `-usage` option to see the possibilities. For example, twelve hours is the default time for the proxy to be in effect. To set the time otherwise, use `grid-proxy-init -hours <nhours>`. While your proxy is in effect, you do not have to enter your passphrase on any Grid machine. If your proxy expires, simply rerun `grid-proxy-init` to create a new proxy.

Destroying a Proxy

To get rid of your proxy when you are finished with it, enter `grid-proxy-destroy`. The proxy will self-destruct after 12 hours (or whatever duration you set), but it is a good idea to destroy it yourself when you are finished.

Determining Your Proxy Status

Use `grid-proxy-info -all` to display the contents or test the status of your proxy.

Chapter 5 Running a Job

In this chapter you will run a few simple Globus jobs, using `globus-job-run` and `globus-job-submit`. Then you will write an RSL (Resource Specification Language) script to use with `globusrun`. Remember that you must have a valid proxy before running Globus. The Globus Resource Allocation Manager (GRAM) authenticates and processes the requests for resources for remote application execution, and allocates the required resources. The commands in this chapter are "GRAM tools."

Commands to Run a Job

globus-job-run

`globus-job-run` is the basic command for running Globus jobs. `globus-job-run` runs in the foreground and defaults to sending output to your terminal. In its basic form, it is roughly equivalent to `rsh`, but also has considerably more functionality for running complex jobs on the Grid.

globus-job-submit

`globus-job-submit` is for submitting jobs to a remote batch job scheduler such as the Portable Batch System (PBS). With `globus-job-submit`, you can submit a job, log out, and log back in later to collect the output. That is, `globus-job-submit` runs in the background and defaults to sending output to the machine running the command.

globusrun

`globusrun` is for submitting jobs that are specified using RSL, the Resource Specification Language. It can run jobs either in the foreground or background, and can send output to your terminal or to the machine running the command. The trend in Globus software development is toward considering `globusrun` as middleware, which can be used by application specific shell scripts to manage job submission. In fact, `globus-job-run` and `globus-job-submit` are simply shell scripts that use `globusrun` for job submission, but present a simpler interface to users.

mpirun

MPICH-G, an implementation of the Message Passing Interface (MPI) standard based on Globus, has an `mpirun` command that can be used to submit MPI parallel jobs to multiple computers in a Grid. See note on page vi, regarding IRIX problem in 1.1.3.

Are You Ready to Run?

Before running a Globus job, try this command to see if you have a current (today's) proxy.

```
% grid-proxy-info   -all
```

In the response from `grid-proxy-info`, your `timeleft` should have some time:

```
timeleft : 11:42:20
```

To see also if your resource is available and you are in the grid-mapfile:

```
% globusrun -a -r <your_resource>
```

For example:

```
% globusrun -a -r evelyn.nas.nasa.gov
```

If you get, "ERROR: resolving resource manager," the Grid Information Services (see page 20) may be malfunctioning.

globus-job-run: Hello World

The basic syntax of `globus-job-run` is the same as `rsh`. This example assumes you have set your path and GLOBUS_INSTALL_PATH environment variable (page 7) so that you can start your command line with a Globus command, and that you have created a proxy using `grid-proxy-init`. The % in the examples indicates the Unix prompt (yours may vary); do not type it. The backslash (\) in long examples means that the following printed line should be a continuation of the current line; DO NOT leave it in when you copy the line.

```
% globus-job-run  evelyn.nas.nasa.gov   /bin/echo  "Hello World."
```

prints:

```
Hello World
```

First comes the Globus command (`globus-job-run`), then the hostname of the machine to run the command on (in this example, evelyn.nas.nasa.gov), then the program to run (the Unix `echo` program), and then the argument (`Hello World`) for the program to use. `Hello World` is the output of the program. When the job is complete `globus-job-run` terminates.

See http://www.globus.org/v1.1/programs/globus-job-run.html for additional examples.

globus-job-run Examples

Using globus-job-run with files—files not provided. These examples show how to use `globus-job-run` with a set of simple hello-world programs that are *not* part of the Globus installation. Substitute your own files and filenames, and substitute your own machine name for denali.mcs.anl.gov. For example, to run the executable /tmp/hw on denali with two arguments that the executable sums up:

```
% globus-job-run  denali.mcs.anl.gov   /tmp/hw  2 5
```

prints:

```
Hello from denali.mcs.anl.gov, sum = 7

% ls -F
```

prints:

```
hw.c hw.o hw.irix*
```

To stage the executable `hw.irix`, which resides on the local workstation, over to the remote machine, execute it, and automatically remove the staged copy after the program has finished:

```
% globus-job-run denali.mcs.anl.gov   -stage  hw.irix 4 6
```

prints:

```
Hello from denali.mcs.anl.gov, sum = 10
```

To start a multi-request with 3 subjobs, the -: delimiter denotes the start of each subjob. If no path is given, the executables are assumed to reside in your home directory on each of the remote hosts. (The executable in this example is compiled with MPICH-G.):

```
% globus-job-run -args 3 5 \
-: evelyn.nas.nasa.gov   -np 2 mpi-hw.irix \
-: pitcairn.mcs.anl.gov   -np 2 mpi-hw.solaris \
-: denali.mcs.anl.gov   mpi-hw.irix 1 2
```

prints:

```
my_id 0 numprocs 5  I am evelyn.nas.nasa.gov, sum = 8
my_id 1 numprocs 5  I am evelyn.nas.nasa.gov, sum = 8
my_id 2 numprocs 5  I am pitcairn.mcs.anl.gov, sum = 8
my_id 3 numprocs 5  I am pitcairn.mcs.anl.gov, sum = 8
my_id 4 numprocs 5  I am denali.mcs.anl.gov, sum = 3
```

Some parameters can be given job-wide (such as the `-args` option). In addition, these parameters can be overridden at a per-subjob level. In the above example, the executable running on denali gets two other arguments to sum up.

Batch Job Submissions

`globus-job-submit` is similar to `globus-job-run`, but is intended for batch job submission. For example, you can remotely submit a job to some local scheduling manager such as PBS, log out, and log back in later to collect the output. `globus-job-submit` defaults to caching the output on the remote machine, for later manual retrieval using the `globus-job-get-output` command. Via a command line interface, you can submit jobs either to a single resource or to a collection of resources. To see the options use `globus-job-submit –help`.

See also http://www.globus.org/v1.1/programs/globus-job-submit.html .

globus-job-submit Example

```
% globus-job-submit  ico16.mcs.anl.gov  -np 32  -maxtime 120  hw.aix
6 8
```

prints:

```
https://ico16.mcs.anl.gov:60106/17916/942265377/
%
```

This example submits a request for 120 minutes of compute time on 32 nodes to the scheduler on the IBM SP at Argonne National Laboratory. The https URL returned by `globus-job-submit` is the job contact string, which uniquely identifies the job. The extra % denotes that this command returns immediately after submission.

`globus-job-submit` caches the output at the remote site. You can override this by specifying `-stdout` and `–stderr`.

New in Version 1.1.3

Old way:

```
% globus-job-run  evelyn.nas.nasa.gov/jobmanager-fork
```

New way:

```
% globus-job-run  evelyn.nas.nasa.gov
```

Resource contact strings, also called "job contact strings," are the part of the Globus command line where you specify which host to use, for example, `evelyn.nas.nasa.gov`. Resource contact strings for "fork" services are no longer routinely suffixed with `/jobmanager-fork`. If you do not specify a jobmanager in your `globusrun`, `globus-job-run`, or `globus-job-submit` command, your job will be submitted to the default jobmanager on the remote system, which is usually fork.

If you want to specify a non-default jobmanager, do so by appending `/jobmanager` and the name of the job scheduler, such as `/jobmanager-lsf` or `/jobmanager-pbs`, to the hostname. For example:

```
% globus-job-run  evelyn.nas.nasa.gov/jobmanager-pbs \
   /bin/echo  "Hello World."
```

Other globus-job-* Commands

To query the status of the job, enter `globus-job-status` followed by the job contact string URL (see paragraph above). For example:

```
% globus-job-status
https://ico16.mcs.anl.gov:60106/17916/942265377/
```

which, for this example, prints:

```
ACTIVE
```

The different states are PENDING (waiting in the queue), ACTIVE (running), SUSPENDED, DONE, and FAILED.

You can retrieve the cached output from the job while the job is running.

```
% globus-job-get-output
https://ico16.mcs.anl.gov:60106/17916/942265377/
```

prints:

```
my_id 0 numprocs 32, sum = 14 : now sleeping for 90 minutes
my_id 1 numprocs 32, sum = 14 : now sleeping for 90 minutes
my_id 2 numprocs 32, sum = 14 : now sleeping for 90 minutes
[...]
```

To cancel a job, enter:

```
% globus-job-cancel https://ico16.mcs.anl.gov:60106/17916/942265377/
```

which prints:

```
Are you sure you want to cancel the job now (Y/N) ? y
Job canceled.
```

```
NOTE: You still need to clean files associated with the job by
running globus-job-clean <jobID>
```

Cached output from the job is not removed. You can still use `globus-job-get-output` to retrieve it after you have cancelled the job.

To cancel a job if it still running and remove the cached output from the remote host, enter:

```
% globus-job-clean https://ico16.mcs.anl.gov:60106/17916/942265377/
```

which prints:

```
WARNING: Cleaning a job means:
- Kill the job if it still running
- Remove the cached output on the remote host
Are you sure you want to cleanup the job now (Y/N) ? y
Cleanup successful.
```

Globus Resource Specification Language

The Globus Resource Specification Language (RSL) is the underlying language that Globus uses to specify resources needed for a Globus job. The `globus-job-run` and `globus-job-submit` commands hide the details of this language from the average user. However, advanced users may want to use RSL directly to handle complicated resource descriptions.

The `globusrun` command supports functionality similar to both `globus-job-run` and `globus-job-submit`, but uses RSL to describe jobs. Normally you write a resource specification in RSL and save it as a file. Then you use `globusrun`, passing it the name of the RSL file as an argument, to run the job described by the RSL. See also page 24.

Writing a Simple RSL Script: Hello Globus World

The RSL allows resources to be specified as a logical expression of <*attribute*, *value*> pairs. Here is a sample RSL that uses the Unix "echo" program to print "Hello Globus World" to your screen. Create a file named hello.rsl and enter the following:

```
&  (count=1)
   (executable=/bin/echo)
   (arguments="Hello Globus World")
```

`&`	Indicates start of RSL script. *&* is used as the first character for GRAM jobs (only one resource). DUROC jobs (multiple resources) start with +.
`(count=1)`	How many processors do you plan to use? (count=4) runs 4 separate processes, and result in 4 Hellos.
`(executable=/bin/echo)`	This calls the normal Unix "echo" command.
`(arguments="Hello Globus World")`	This supplies the argument to echo.

RSL Syntax

See http://www.globus.org/gram/rsl_spec1.html for a complete description of RSL syntax.

Letting globus-job-run Write the RSL

You can get `globus-job-run` to write your RSL. Use the `globus-job-run` command as above, with `-dumprsl` as the first option.

```
% globus-job-run  -dumprsl evelyn.nas.nasa.gov   /bin/echo  \
"Hello Globus World"
```

prints:

```
&(executable="/bin/echo")
   (arguments= "Hello Globus World")
```

Save the two-line output as `hello.rsl`

Using the globusrun Command

To run the `hello.rsl` script shown above, use `globusrun`.

```
% globusrun -s -r <host name> -f hello.rsl
```

For example, after the Unix prompt, enter `globusrun` with the `-s` option to send output to stdout. Use the `-r` option to indicate that a resource name follows. The `-f` option indicates a filename follows, then your filename.

```
% globusrun -s -r evelyn.nas.nasa.gov   -f hello.rsl
```

“Hello Globus World” ought to appear on your screen. Are you having problems? Did you use `grid-proxy-init`? The `globusrun` command requires the existence of a valid proxy to function correctly.

Optionally, a resource name takes the form `<host name>/jobmanager-<scheduler type>` if you want to specify a service other than the default, which is usually fork. For example, a service name of “`jobmanager-pbs`” would be used to submit a job to a PBS scheduler. See page 15.

`globusrun` takes your RSL script and parcels its instructions out to specified resources. The RSL script by itself is a resource specification, not an executable.

In addition to starting jobs, `globusrun` can be used to list previously started jobs, query the status of previously started jobs, parse RSL request strings, and perform authentication tests to GRAM gatekeepers. In fact, the `globus-job-run` and `globus-job-submit` programs simply use `globusrun` to implement much of their functionality.

More globusrun Examples

Example One

```
% globusrun -s -r pitcairn.mcs.anl.gov  '&(executable=my_prog)'
```

Resolve the contact string for the fork jobmanager service on host pitcairn, and then submit the program "my_prog" to that resource. The standard output and standard error of `my_prog` will be displayed by `globusrun`.

Example Two

```
% globusrun -s -r ico16.mcs.anl.gov/jobmanager-easymcs \
'&(executable=$(GLOBUSRUN_GASS_URL)/usr/local/my_prog) \
(stdout=/tmp/output1)(stderr=/tmp/error1)(count=3)'
```

Submit three copies of the local program /usr/local/my_prog to the EASY scheduler at ico16.mcs.anl.gov, and send the output to the remote /tmp directory.

Example Three

```
% globusrun  -l | awk '/my_prog/ {print $1}'
```

List the job IDs of jobs submitted that contain the string my_prog.

Example Four

```
% globusrun -w -r pitcairn.mcs.anl.gov \
'&(executable="$(GLOBUS_TOOLS_PATH)/bin/globus-url-copy") \
(arguments="file:/tmp/myout.1" $(GLOBUSRUN_GASS_URL)/tmp/myout.1)
```

Copy a file from pitcairn to the local host, using the GASS transfer protocol. The `-w` option gives the remote process write permission on the local GASS server started by `globusrun`. Note that because the colon (:) is a reserved RSL character, the first argument must be quoted. See http://www.globus.org/gram/rsl_spec1.html for more information.

Checking and Killing Jobs

What is My Job ID?

Enter `globusrun -list` or `globusrun –l` to get your job ID.

Is My Job Running?

Use `globus-job-status` or `globusrun –list` to check if your job is running. In general, if your Unix prompt hasn't returned after a `globus-job-run` command, your job is still running. You can log on to the remote resource you specified and use a Unix command like `ps`.

Killing a Job

If you interrupt the `globusrun` process (by entering CTRL-C or sending it a SIGINT), your jobs should automatically be canceled.

The job cancellation routines (triggered by CTRL-C or SIGINT in the `globusrun` process) have a timeout to allow an intelligent job to cleanup. Therefore, you should expect some delay before the `globusrun` process exits.

What Happens When You Submit a Job

This is a complicated process, but in streamlined form it looks like this:

1. You run `globusrun` with an RSL script (or `globus-job-run` or `globus-job-submit`) and specify where to run.

2. `globusrun` contacts something called a "gatekeeper" on the remote host and performs mutual authentication. This means that the remote host knows who you are, and you know who the remote host is.

3. The gatekeeper (on the remote host) contacts a job manager service with your request. The job manager will decide how to run your job depending on the service name that is used. If you did not specify a service, the default service is used, usually “fork,” which will run the job immediately. If you specified a job scheduler (e.g., /`jobmanager-pbs`), your job will be submitted to the scheduler and will run as the scheduler allows. See page 15.

4. Your job completes.

Global Resource Allocation Manager

The Globus Resource Allocation Manager (GRAM) authenticates and processes the requests for resources for remote application execution, and allocates the required resources. It also prints updated information regarding the capabilities and availability of computing resources to the Grid Information Service (see page 20).

GRAM provides an API for submitting and canceling a job request, as well as checking the status of a submitted job. You write the specifications in the Resource Specification Language (RSL). The specifications are processed by GRAM as part of the job request.

Your program can call the GRAM C API directly and supply it with an RSL, or you can use `globusrun`, giving it the RSL, as in the Hello World example:

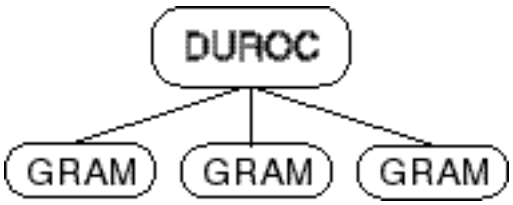
```
% globusrun  -s -r evelyn.nas.nasa.gov   -f  hello.rsl
```

GRAM manages jobs that use only one resource. For more information, see http://www.globus.org/gram For a tutorial, see http://www.globus.org/gram/tutorial_gssapi_ssleay.html

The commands in this chapter are “GRAM tools.”

Dynamically Updated Request Online Co-allocator

Dynamically Updated Request Online Co-allocator (DUROC) co-allocates multiple resources and manages multiple GRAM jobs. See http://www.globus.org/duroc for more information.



Chapter 6 Grid Information Service

The Grid Information Service (GIS) stores information about the state of the Grid infrastructure. GIS is a new term for what was previously known as Metacomputing Directory Service (MDS). The term “MDS” is still widely used. Use the Globus `grid-info-*` commands to gather information from the GIS. GIS is work in progress; there are still problems to resolve. This chapter does not address initialization or population of information into the GIS.

Terminology

GIS is a service that allows the storage of information about the state of the Grid infrastructure. One of its services is to publish information via LDAP (Lightweight Directory Access Protocol), a protocol used to locate resources in a network.

The GIS information service has the ability to function as a white pages directory—for retrieving information associated with a particular name ("distinguished name"). Examples for such lookups are the number of CPUs and the operating system associated with a particular machine. The GIS also functions as a yellow pages directory—for retrieving a list of categorized entities. Such categories are defined by "object classes." Examples of such categories are lists of computers or people.

A powerful extension to the white and yellow pages function of the directory is the ability to augment the lookups with sophisticated Boolean search filters.

The most important change in the GIS/MDS from 1.1.2 to 1.1.3 is that the Globus group no longer runs a centralized MDS server. That means that each site optionally will have its own organization server, which is used for any searching commands.

Globus version 1.1.3 introduces two subsets of GIS: the Grid Resource Information Service (GRIS) and the optional Grid Information Index Service (GIIS). The new GIS model is push rather than pull, and will cache information from resources within a particular organization. For more information, see: http://www.globus.org/mds, and http://www.globus.org/toolkit/download/info-113.html

Tools

See http://www.globus.org/mds for GIS search tools other than the `grid-info-*` commands.

grid-info-search

The command `grid-info-search` allows searches on the GIS server based on search filters that conform to LDAP searches . The format is:

```
grid-info-search [ options ] <search filter> [attributes ]
```

To see the options available, and find out your host and port information, type:

```
% grid-info-search  -help
```

To list all distinguished names along with all attributes of the compute resources that are stored in the directory under the local organization:

```
% grid-info-search "(objectclass=GlobusComputeResource)"
```

To view information from other organizations, you must know the host and port of the GIIS server at that organization, and include the `-mdshost` or `-h` <host>, `-mdsport` or `-p` <port>, and `-b` <base> options to `grid-info-search`. For example:

```
% grid-info-search -h mds.nas.nasa.gov -p 389 -b "o=globus,c=us"
"(objectclass=GlobusComputeResource)"
```

To list all distinguished names of the compute resources that are stored under the local organization, along with the operating system type of each:

```
% grid-info-search  "(objectclass=GlobusComputeResource)" dn ostype
```

To restrict the attributes that are output, list them at the end of the `grid-info-search` command. Search filters are specified in Polish notation where *&* is the Boolean *and* operator and | is the Boolean *or* operator. It is a good practice to restrict the searches to objectclasses of interest in order to minimize the duration it takes for a query to return. As a general rule of thumb, the more precise the query, the shorter the response time. Tip: you also can use the Unix `grep` command to narrow your `grid-info-search` results. For example, to list hostnames:

```
% grid-info-search "(Objectclass=*)" | grep "objectname=service"
```

The new org DN namespace is the "domain component" explosion of the site's DNS domain under the "o=Grid" root; e.g., mcs.anl.gov becomes "dc=mcs, dc=anl, dc=gov, o=Grid" or isi.edu becomes "dc=isi, dc=edu, o=Grid".

Chapter 7 Accessing Remote Data

Globus Access to Secondary Storage (GASS) provides programs and C APIs for remotely accessing data. This chapter describes how to start a GASS server, and how to use the `globus-url-copy` to transfer data to and from GASS servers. GASS programs allow data to be easily transferred from one machine to another.

GASS Options

GASS provides programs and C APIs for remotely accessing data. Several commands and programs support secondary storage functionality.

globus-rcp

`globus-rcp` is the recommended command for remote file transfer. The `globus-rcp` is the functional equivalent of the Unix `rcp` program, but uses GASS to perform its remote data transfer. You do not need to start a `globus-gass-server` by hand to use `globus-rcp`. Instead, `globus-rcp` will use GRAM to automatically start and stop remote GASS servers and clients for you. An example:

```
% globus-rcp evelyn.nas.nasa.gov:/tmp/foo /tmp/bar
```

Here the /tmp/foo file on evelyn is copied to the /tmp/bar on the machine running this command.

globus-gass-server

The `globus-gass-server` program is used to make the data on one computer available to remote clients. To run, enter:

```
% globus-gass-server
```

prints

```
https://evelyn.nas.nasa.gov:20143
```

This sets up your machine to serve files to remote clients. The https URL that is output by the GASS server contains the host and port on which this server is listening for requests. It is used by clients to transfer data to and from this machine via the GASS server.

globus-url-copy

The `globus-url-copy` command is used to remotely access files from GASS servers, or from other servers that speak the http or https protocols (e.g., a web server). The general form of `globus-url-copy` is:

```
% globus-url-copy <fromURL> <toURL>
```

Examples:

```
% globus-url-copy https://evelyn.nas.nasa.gov:20143/tmp/foo
file:/tmp/bar
% globus-url-copy https://evelyn.nas.nasa.gov:20143/tmp/foo -
```

In these examples, the <*fromURL*> is composed of the base URL that was returned by the `globus-gass-server` (`https://evelyn.nas.nasa.gov:20143`), followed by the path of a file accessible to `globus-gass-server` (/tmp/foo). In the first example, the /tmp/foo file is remotely transferred from evelyn to a local file named /tmp/bar (file:/tmp/bar). In the second example, the single dash (`-`) redirects the contents of the remote /tmp/foo file to stdout.

Chapter 8 RSL With Shell Script

Included in this chapter is a Resource Specification Language (RSL) file that calls a shell script that will be submitted by Globus to PBS (Portable Batch System), and a Bourne shell script, called by the RSL, that runs an application, OVERFLOW.

The RSL File

```
&       (count=4)
        (maxTime=20)
        (jobType=single)
        (directory=/scratch1/yarrow/X38.globus/)
        (executable=$(GLOBUS_TOOLS_PATH)/bin/globus-sh-exec)
        (arguments=run.sh)
        (stdout=run.out)
        (stderr=run.err)
        (environment=(GASS_URL $(GLOBUSRUN_GASS_URL)
               (OSNAME $(GLOBUS_HOST_OSNAME))
```

The executable here refers to a shell script that will be wrapped to be a PBS script.

The Shell Script

```
sourcefile=/scratch1/yarrow/X38-repository/all.tar
execprog=/ipg-home/yarrow/Overflow2.0.7/overflow2d.${OSNAME}

globus-url-copy ${GASS_URL}${sourcefile} - | tar xf -
globus-url-copy ${GASS_URL}${execprog} - | cat > overflow-exec
mpirun -np 4 overflow-exec
```

The shell script is getting `mpirun` to start Overflow2 (an Overflow variant) as a four-processor parallel job.

Chapter 9 Globus Commands

This chapter shows you the commands available in Globus 1.1.x. On your Unix command line, type a Globus command followed by the `-usage` option to get help. Effective with version 1.1 of, the first word of some of the commands has been changed from `globus` to `grid`. You will find information on the complete set of command-line programs in the Globus Toolkit at http://www.globus.org/v1.1/programs.

Also refer to the command information in Chapter 5.

Globus 1.1.3 Tools

The following tools are available in <*your-globus-tools-path*>in the Globus Toolkit version 1.1. For usage information, including options available, type the command name with the `-usage` option. See also http://www.globus.org/v1.1/programs.

Security

`grid-cert-request`	Creates a new certificate request and private key.
`grid-cert-info`	Displays certificate information.
`grid-cert-renew`	Creates a new key and renewal request for a Globus certificate.
`grid-change-pass-phrase`	Changes the pass phrase that protects your private key.
`grid-proxy-init`	Creates a proxy certificate that can be used for authentication without having to reenter the protecting pass phrase for each resource.
`grid-proxy-info`	Displays proxy certificate information.
`grid-proxy-destroy`	Removes your proxy certificate.

Job Submission

`globusrun`	Run a single executable on a remote site .
`globus-setup-test`	Verifies your setup of credentials.
`globus-job-cancel`	Cancels a job previously started using globus-job-submit.

globus-job-run	Allows you to run a job at one or several remote resources. It translates the program arguments to a RSL request and uses globusrun to submit the job.
globus-job-clean	Kills the job if it is still running and cleans the information concerning the job.
globus-job-get-output	For the job specified, gets the standard output or standard error resulting from the job execution.
globus-job-status	Display the status of the job. See also globus-get-output to check the standard output or standard error of your job.
globus-job-submit	For batch job submission (i.e., submitting a job to a queue via some local scheduling manager like PBS).

Information Services

grid-info-add	Modifies the GIS server based on the contents of input file.
grid-info-host-search	Searches the GIS on a specified machine.
grid-info-remove	See grid-info-add.
grid-info-search	Searches the GIS.
grid-info-site	Searches all machines associated with a given site.
grid-info-update	See grid-info-add.

Other Tools

globus-hostname	This is a simple shell script that acts like the Unix hostname.
globus-hostname2contacts	Converts a hostname to a list of resource manager contact strings.
globus-netstat	Hides the implementation-specifics of netstat and reformats the output to be consistent across architectures, producing a subset of UNIX System V netstat output.
globus-sh-exec	Sources the globus-sh-tools file, then executes a user script.
globus-version	Shows version number.
globus-development-path	Prints the full path to the "development" directory (include files and libraries) that corresponds best to the flavor indicated by the command-line options (pthreads, debug, 64-bit, support for SHM, ...)
globus-install-path	Prints the full path to the Globus install tree.

`globus-rcp`	Remote copies using GASS and Globus submission. Many options.
`globus-tools-path`	Prints the full path to the tools directory in the Globus install tree, tailored for the current architecture.
`globus-services-path`	Prints the full path to the services directory in the Globus install tree, tailored for the current architecture.
`globus-tilde-expand`	Expands the leading tilde sign (~) (and the specified username if provided) to the full path of your home directory.

Chapter 10 Bibliography and Reference

Web and Email Resources

- http://www.globus.org
  the central site
- http://www.globus.org/about/contacts.html
  Problem Report Form and list of contacts
- http://www.globus.org/about/faq.html Frequently Asked Questions.
- discuss@globus.org
  to ask question or discuss approaches in developing grid-aware applications via the globus tool kit. Subscribe by sending mail to majordomo@globus.org with `subscribe discuss` in the body of the message
- http://www.nas.nasa.gov/Software/p2d2/
  NAS debugging system for parallel and distributed programs
- documentation@globus.org
  report errors in this document

Technical Papers

Proc. 8th IEEE Symposium on High-Performance Distributed Computing, "Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid", William E. Johnston, Dennis Gannon and Bill Nitzberg, 1999, HPDC8

See http://www.globus.org/research/papers.html for a large selection of papers.

The Grid Book

The Grid: Blueprint for a New Computing Infrastructure, Edited by Ian Foster and Carl Kesselman: http://www.mkp.com/books_catalog/1-55860-475-8.asp

Chapter 11 Glossary

DUROC

Dynamically Updated Request Online Co-allocator

GASS

Global Access to Secondary Storage (remote file management)

GIS

Grid Information Service (formerly MDS, Metacomputing Directory Service) for locating and distributing characteristics of resources. Accesses the white pages and yellow pages served by LDAP. Used with a set of commands each beginning with `grid-info`. See http://www.globus.org/mds

Globus

The Globus Project is a community effort, led by Argonne National Laboratory and the University of Southern California's Information Sciences Institute. Globus is developing the basic software infrastructure for computations that integrate geographically distributed computational and information resources. http://www.globus.org

GRAM

Globus Resource Allocation Manager
http://www.globus.org/gram.

Grids

Widely distributed networks of high-performance computers, stored data, instruments, and collaboration environments.
http://www.globus.org/testbeds.

HBM

HeartBeat Monitor that checks if a process is alive. The Globus Heartbeat Monitor (HBM) is designed to provide a simple, highly reliable mechanism for monitoring the state of processes. The HBM is designed to detect and report the failure of processes that have identified themselves to the HBM.
http://www.globus.org/cgi/hbm-simple.cgi.

help

Type any command with the `-usage` or `-help` option for online help.

LDAP

Lightweight Directory Access Protocol, for accessing information about hardware, software, and status in a networked environment. Directory-server software available from numerous sources. http://www.globus.org/mds

Maxtime

Maximum time a job should be allowed to run. Refers to wallclock time on some systems, CPU time on others.

MDS

Metacomputing Directory Service, the former name of GIS. http://www.globus.org/mds

MPI

Message Passing Interface, the industry-wide standard protocol for passing messages between parallel processors.

MPICH

A portable MPI model implementation.

MPICH-G

Message Passing Interface for wide-area networks; the Globus version of MPI. See http://www.globus.org/v1.1/software/mpich.html

PBS

Flexible batch processing software that operates on networked, multi-platform Unix environments, including heterogeneous clusters of workstations, supercomputers, and massively parallel systems.
http://pbs.mrj.com.

resources

Computers, instruments, and immersive environments; machines.

RSL

Resource Specification Language. A language that can be used with tools such as `globusrun` to specify resource requirements, such as what executable(s) to use, arguments to pass, etc.

usage

Type any command with the `-usage` or `-help` option for online help.

Chapter 12 Index